

4. 代数(多項式)方程式および超越関数方程式を解く

- 代数(多項式)方程式
- ユーザー定義による関数
- 超越関数方程式($\cos(x)$, $\sin(x)$, $\log(x)$)
- Symbolic computation(文字式の計算)

線形方程式を解く： roots

- 2階の代数方程式“ $x^2-x-2=(x-2)(x+1)=0$ ”を解く：
この方程式は $\mathbf{X}=(x^2,x^1,x^0)^T$ と $\mathbf{C}=(1,-1,-2)$ を用いて $\mathbf{CX}=\mathbf{0}$ と記述できる。
この行列 \mathbf{C} を用いて以下の様に解くことができる。

```
>> C=[1,-1,-2];  
>> roots(C)  
ans =  
     2  
    -1
```

- 3階の代数方程式“ $x^3+1=0$ ”を解く：
同様に $\mathbf{X}=(x^3,x^2,x^1,x^0)^T$ と $\mathbf{C}=(1,0,0,1)$ を用いて $\mathbf{CX}=\mathbf{0}$ と記述できる。
この行列 \mathbf{C} を用いて以下の様に解くことができる。

```
>> C=[1,0,0,1];  
>> roots(C)  
ans =  
-1.00000 + 0.00000i  
 0.50000 + 0.86603i  
 0.50000 - 0.86603i
```

ユーザー定義の関数

- スクリプトフォームに次のように記述することで、任意の関数を定義することができる。

```
function [y1, ..., yN] = myfun(x1, ..., xM)
y1 = ...
...
end
```

- スクリプトに次のように入力して、“myfun.m”の名前で保存:

```
#myfun.m
function y = myfun(x)
y = x^2+sin(x)-1;
end
```

- 次の様に関数を呼び出して使用することが可能:

```
>> myfun(0)
ans = -1
>> myfun(1)
ans = 0.84147
```

Remark: These commands must be run in the same directory (folder) as myfun.m was saved. Or you can add the directory where myfun.m exists to Octave's load path; type “help path” for details.

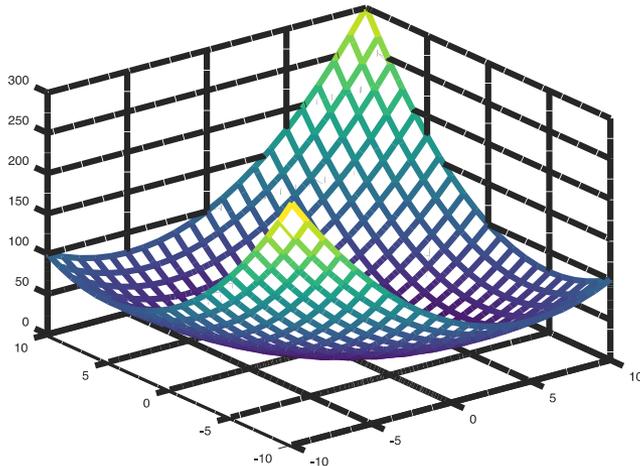
スクリプトを使わずに演算(無名関数)

- スクリプトで関数を定義しなくても同様の演算は可能

```
>> myfun1 = @(x) (x^2+sin(x)-1);  
>> myfun1(1)  
ans = 0.84147
```

- 二変数関数の例:

```
>> myfun2 = @(x,y) (x.^2+y.^2+x.*y);  
>> [X,Y] = meshgrid(-10:10);  
>> mesh(X,Y,myfun2(X,Y))
```



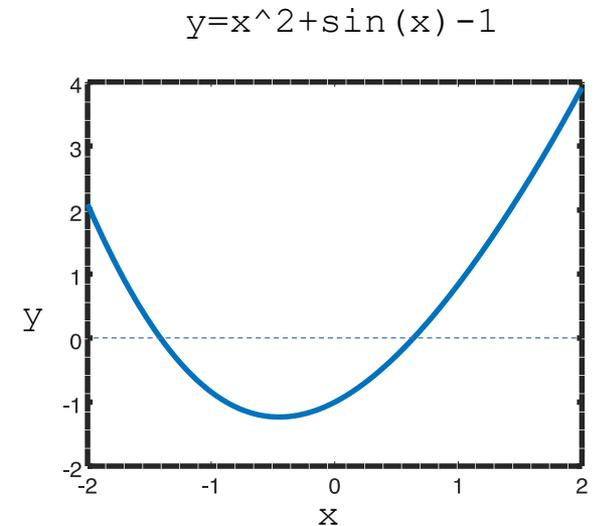
Remark: The use of `x.^2` instead of `x^2` above makes it possible to deal with the case when `x` is a matrix (or a vector or even a tensor).

超越関数方程式を解く：fsolve

- To find roots of $x^2 + \sin(x) - 1 = 0$, type as follows:

```
>> fsolve(@(x) x^2+sin(x)-1, 1.0)  
ans = 0.63673 初期値  
>> fsolve(@(x) x^2+sin(x)-1, -1.0)  
ans = -1.4096 初期値
```

- “fsolve”は与えられた初期値を基に解を探します。
- 解を得られるかどうかは与えられた方程式と初期値に依ります。



20.1 Solvers From <https://www.gnu.org/software/octave/doc/>

Octave can solve sets of nonlinear equations of the form

$$F(x) = 0$$

using the function `fsolve`, which is based on the MINPACK subroutine `hybrd`. This is an iterative technique so a starting point must be provided. This also has the consequence that convergence is not guaranteed even if a solution exists.

Function File: `fsolve` (*fcn*, *x0*, *options*)

Function File: [*x*, *fvec*, *info*, *output*, *fnjac*] = `fsolve` (*fcn*, ...)

Solve a system of nonlinear equations defined by the function *fcn*.

fcn should accept a vector (array) defining the unknown variables, and return a vector of left-hand sides of the equations. Right-hand sides are defined to be zeros. In other words, this function attempts to determine a vector *x* such that *fcn* (*x*) gives (approximately) all zeros.

x0 determines a starting guess. The shape of *x0* is preserved in all calls to *fcn*, but otherwise it is treated as a column vector.

Symbolic package のインストール

- 以下のページに実習で使うSymbolic package に関する詳細があるので適宜参照

<https://github.com/cbm755/octsympy>

パッケージのインストールは次のページから

“symbolic-win-py-bundle-2.6.0.zip”をダウンロードして作業ディレクトリに保存

<https://github.com/cbm755/octsympy/releases>

- 以下のコマンドをコマンドラインに入力:

```
>> pkg install symbolic-win-py-bundle-2.6.0.zip
>> pkg load symbolic
```

- シンボル計算を行うにはまず“syms”コマンドを使ってシンボル変数を定義する

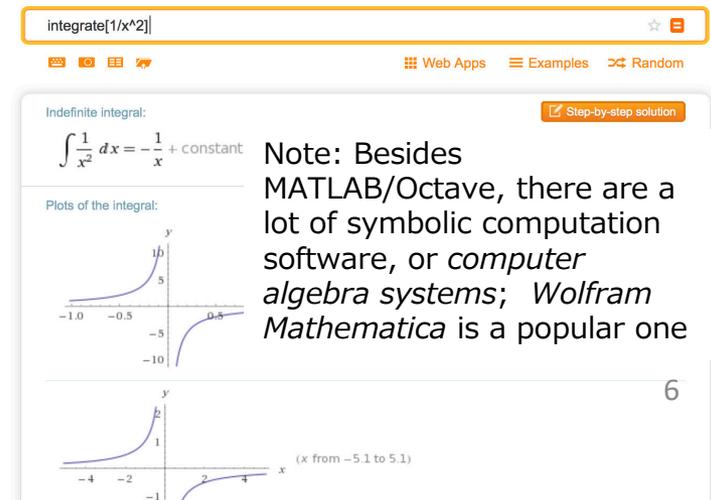
```
>> syms x
```

- 関数のシンボル表現:

```
>> x^2+sin(x)-1
ans = (sym)
      2
      x  + sin(x) - 1
```

<http://www.wolframalpha.com>

 **WolframAlpha** computational knowledge engine.



The screenshot shows the WolframAlpha interface. The search bar contains the query "integrate[1/x^2]". Below the search bar, the results are displayed. On the left, there is a plot of the function $y = \frac{1}{x^2}$ for x from -1.0 to 1.0. The plot shows a vertical asymptote at $x = 0$ and the curve approaching the x-axis as $|x|$ increases. On the right, the text reads: "Indefinite integral: $\int \frac{1}{x^2} dx = -\frac{1}{x} + \text{constant}$ ". Below this, there is a note: "Note: Besides MATLAB/Octave, there are a lot of symbolic computation software, or computer algebra systems; Wolfram Mathematica is a popular one". At the bottom right of the screenshot, the number "6" is visible.

Symbolic package: 因数分解

- 多項式の因数分解: factor

```
>> syms x
>> f=x^3+13*x^2-105*x+171;
>> factor(f)
ans = (sym)
      2
      (x - 3) * (x + 19)
```

```
>> syms x y
>> f=x^3*y-3*x^3-4*x^2*y+12*x^2-3*x*y+9*x+18*y-54;
>> factor(f)
ans = (sym)
      2
      (x - 3) * (x + 2) * (y - 3)
```

Symbolic package: 微分

- シンボル表記された数式の微分: diff

```
>> diff(x^2+sin(x)-1)
ans = (sym) 2*x + cos(x)
```

```
>> diff(exp(-x*sin(x)))
ans = (sym)
(-x*cos(x) - sin(x))*e-x*sin(x)
```

Symbolic package: 不定積分

- シンボル表記された数式の積分 : int

```
>> int(x^2+sin(x)-1)
ans = (sym)
```

$$\frac{x^3}{3} - x - \cos(x)$$

```
>> int(sin(log(x)))
ans = (sym)
```

$$\frac{x \cdot \sin(\log(x))}{2} - \frac{x \cdot \cos(\log(x))}{2}$$

Exercises 4.1

- 次の方程式の解を全て求めよ

$$(\sin x)^2 \exp\left(-\frac{x}{2}\right) - 0.1 = 0, \quad (x \geq 0)$$

- ヒント: “fsolve”で解くためには適当な初期値を用いる必要があります。
そこで、以下のコマンドで $y=f(x)$ をプロットすることで適当な初期値を検討してみてください。

```
>> x=0:0.01:10;  
>> y=sin(x).^2.*exp(-x/2) - 0.1;  
>> y0=zeros(1,length(x));  
>> plot(x,y,x,y0)
```

提出先：
東北大学インターネットスクール(ISTU)上で提出
もしくは
Email: hisashi.kino.a1@tohoku.ac.jp
shimada@m.tohoku.ac.jp

〆切：
2018年7月2日の午前8:50